Chained Gaussian Processes

Alan D. Saul
Department of Computer Science
University of Sheffield

James Hensman
CHICAS, Faculty of Health and Medicine
Lancaster University

Aki Vehtari
Helsinki Institute for Information Technology HIIT
Department of Computer Science
Aalto University

Neil D. Lawrence
Department of Computer Science
University of Sheffield

Abstract

Gaussian process models are flexible, Bayesian non-parametric approaches to regression. Properties of multivariate Gaussians mean that they can be combined linearly in the manner of additive models and via a link function (like in generalized linear models) to handle non-Gaussian data. However, the link function formalism is restrictive, link functions are always invertible and must convert a parameter of interest to an linear combination of the underlying processes. There are many likelihoods and models where a non-linear combination is more appropriate. We term these more general models *Chained Gaussian Processes*: the transformation of the GPs to the likelihood parameters will not generally be invertible, and that implies that linearisation would only be possible with multiple (localized) links, i.e a chain. We develop an approximate inference procedure for Chained GPs that is scalable and applicable to any factorized likelihood. We demonstrate the approximation on a range of likelihood functions.

1 Introduction

Gaussian process models are flexible distributions that can provide priors over non linear functions. They rely on properties of the multivariate Gaussian for their tractability and their non-parametric nature. In particular, the sum of two functions, drawn from a Gaussian process is also given by a Gaussian process. Mathematically, if $f \sim \mathcal{N}(\mu_f, k_f)$ and $g \sim \mathcal{N}(\mu_g, k_f)$ and we define $y = g + f$ then properties of multivariate Gaussian give us that $y \sim \mathcal{N}(\mu_f + \mu_g, k_f + k_g)$ where μ_f and μ_g are deterministic functions of a single input, k_f and k_g are deterministic, positive semi definite functions of two inputs and y, g and f are stochastic processes.

This elementary property of the Gaussian process is the foundation of much of its power. It makes additive models trivial, and means we can easily combine any process with Gaussian noise. Naturally, it can be applied recursively, and covariance functions can be designed to reflect the underlying structure of the problem at hand (e.g. Hensman et al. [2013b] uses additive structure to account for variation in replicate behavior in gene expression).

In practice observations are often *non Gaussian*. In response, statistics has developed the field of *generalized linear models* [Nelder and Wedderburn, 1972]. In a generalized linear model a link function is used to connect the mean function of the Gaussian process with the mean function of another distribution of interest. For example, the log link can be used to relate the rate in a Poisson distribution with our GP, $\log \lambda = f + g$. Or for classification the logistic distribution can be used to represent the mean probability of positive outcome, $\log \frac{p}{1-p} = f + g$.

Writing models in terms of the link function captures the linear nature of the underlying model, but it is somewhat against the probabilistic approach to modeling where we consider the *generative model* of our data. While there's nothing wrong with this relationship mathematically, when we consider the generative model we never apply the link function directly, we consider the inverse link or *transformation function*. For the log link this turns into the exponential, $\lambda = \exp(f + g)$. Writing the model in this form emphasizes the importance that the transformation function has on the generative model (see e.g. work on warped GPs [Snelson et al., 2004]). The log link implies a multiplicative combination of f and g, $\lambda = \exp(f + g) = \exp(f)\exp(g)$, but in some cases we might wish to consider an additive model, $\lambda = \exp(f) + \exp(g)$. Such a model no longer falls within the class of generalized linear models as there is no link function that renders the two underlying processes additive Gaussian. In this paper we address this issue and use variational approximations to develop a frame-

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

work for *non-linear combination* of the latent processes. Because these models cannot be written in the form a single link function we call this approach "Chained Gaussian Processes".

2 Background

Assume we have access to a training dataset of n input-output observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, y_i is assumed to be a noisy realisation of an underlying latent function $\mathbf{f} = f(\mathbf{x})$, i.e. $y_i = f(\mathbf{x}_i) + \epsilon_i$. For a Gaussian likelihood $\epsilon_i \sim \mathcal{N}\left(\mu, \sigma^2\right)$, $\mathbf{x}_i \in \mathbb{R}^q$ and $y_i \in \mathbb{R}$. Normally the mean of the likelihood is assumed to be input dependent and given a GP prior $\mu = \mathbf{f}_i = f(\mathbf{x}_i)$ where $f(\mathbf{x}) \sim \mathcal{GP}(\boldsymbol{\mu}_f, k_g(\mathbf{x}, \mathbf{x}'))$, and σ is fixed at an optimal point. In this case the integrals required to infer a posterior, $p(\mathbf{f}|\mathbf{y})$, are tractable.

One extension of this model is the heteroscedastic GP regression model [Goldberg et al., 1998, Lazaro-Gredilla and Titsias, 2011], where the noise variance σ is dependent on the input. The noise variance can be assigned a log GP prior, $y_i \sim \mathcal{N}\left(f(\mathbf{x}_i), e^{g(\mathbf{x}_i)}\right)$, where $g(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_g, k_g(\mathbf{x}, \mathbf{x}'))$, i.e. a log link function is used. Unfortunately this generalization of the original Gaussian process model is not analytically tractable and requires an approximation to be made. Suggested approximations include MCMC [Goldberg et al., 1998], variational inference [Lazaro-Gredilla and Titsias, 2011], Laplace approximation [Vanhatalo et al., 2013] and expectation propagation [Hernández-Lobato et al., 2014] (EP).

Another generalization of the standard GP is to vary the scale of the process as a function of the inputs. Adams and Stegle [2008] suggest a log GP prior for the scale of the process giving rise to non-parametric non-stationarity in the model. Turner and Sahani [2011] took a related approach to develop probabilistic amplitude demodulation, here the amplitude (or scale) of the process was given by a Gaussian process with a link function given by $\sigma = \log(\exp(f) - 1)$. Finally Tolvanen et al. [2014] assign both the noise variance and the scale a log GP prior.

Both these two variations on Gaussian process regression combine processes in a non-linear way within a Gaussian likelihood, but the idea may be further generalized to systems that deal with non-Gaussian observation noise.

In this paper we describe a general approach to combining processes in a non-linear way. We assume that the likelihood factorizes across the data, but is a general non-linear function of b input dependent latent functions. Our main focus will be examples of likelihoods with $b = 2$, $f(\mathbf{x}_i)$ and $g(\mathbf{x}_i)$, such that, $p(\mathbf{y}|f(\mathbf{x}_i), g(\mathbf{x}_i))$, though the ideas can all be generalized to $b > 2$. Previous work in this domain include the use of the Laplace approximation [Vanhatalo et al., 2013], however this method scales poorly, $\mathcal{O}(b^3 n^3)$ and so isn't applicable to datasets of a moderate size.

To render the model tractable we extend recent advances in large scale variational inference approaches to GPs [Hensman et al., 2013a]. With non-Gaussian likelihoods restrictions on the latent function values may differ, and a non-linear transformation of the latent function, $\mathbf{g} \in \mathbb{R}^q$ may be required. The inference approach builds on work by Hensman et al. [2015], that in turn builds on the variational inference method proposed by Opper and Archambeau [2009].

In other work [Nguyen and Bonilla, 2014] mixtures of Gaussian latent functions have also been applied for non-Gaussian likelihoods, we expect such mixture distributions would also be applicable to our case. More recently this approach [Dezfouli and Bonilla, 2015] was extended to provide scalability using sparse methods similar those used in this work.

3 Chained Gaussian Processes

Our approach to approximate inference in chained GPs builds on previous work in inducing point methods for sparse approximations of GPs [Snelson and Ghahramani, 2006, Titsias, 2009, Hensman et al., 2015, 2013a]. Inducing point methods introduce m 'pseudo inputs', known as inducing inputs, at locations $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$. The corresponding function values are given by $\mathbf{u}_i = f(\mathbf{z}_i)$. These inducing inputs points do not effect the marginal of $\mathbf{f}$ because

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Z}) = \int p(\mathbf{f}|\mathbf{u}, \mathbf{X}) p(\mathbf{u}|\mathbf{Z}) d\mathbf{u},$$

where $p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}(\mathbf{u}|0, \mathbf{K_{uu}})$ and $p(\mathbf{f}|\mathbf{u}, \mathbf{X}) = \mathcal{N}\left(\mathbf{f}|\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\right)$. The part-covariances given by $\mathbf{K_{fu}} = k_f(\mathbf{X}, \mathbf{Z})$ where $\mathbf{X}$ is the locations of $\mathbf{f}$, define the relationship between inducing variables and the latent function of interest, f. The marginal likelihood is $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{f}\, d\mathbf{u}$. To avoid $\mathcal{O}(n^3)$ computation complexity Titsias [2009] invokes Jensen's inequality to obtain a lower bound on the marginal likelihood $\log p(\mathbf{y})$, an approach known as *variational compression*. This approximation also forms the basis of our approach for non-Gaussian models.

3.1 Variational Bound

For non-Gaussian likelihoods, even with a single latent process the marginal likelihood, $p(\mathbf{y})$, is not tractable, but it can be lower bounded variationally. We assume that the latent functions, $\mathbf{f} = f(\mathbf{x})$ and $\mathbf{g} = g(\mathbf{x})$ are *a priori* independent

$$p(\mathbf{f}, \mathbf{g}|\mathbf{u}_f, \mathbf{u}_g) = p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{g}|\mathbf{u}_g). \tag{1}$$

The derivation of the variational lower bound then follows a similar form as [Hensman et al., 2015] with the extension to multiple latent functions. We begin by writing down our

log marginal likelihood,

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\mathbf{f},\mathbf{g})p(\mathbf{f},\mathbf{g}|\mathbf{u}_f,\mathbf{u}_g) \times p(\mathbf{u}_f)p(\mathbf{u}_g)d\mathbf{f}\, d\mathbf{g}\, d\mathbf{u}_f\, d\mathbf{u}_g$$

then introduce a variational approximation to the posterior,

$$p(\mathbf{f},\mathbf{g},\mathbf{u}_g,\mathbf{u}_f|\mathbf{y}) \approx p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g)q(\mathbf{u}_f)q(\mathbf{u}_g), \quad (2)$$

where we have made the additional assumption that the latent functions factorize in the variational posterior.

Using Jensen's inequality and the factorization of the latent functions (1), a variational lower bound can then be obtained for the log marginal likelihood,

$$\begin{aligned}\log p(\mathbf{y}) &= \log \int p(\mathbf{y}|\mathbf{f},\mathbf{g})p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g) \times p(\mathbf{u}_f)p(\mathbf{u}_g)d\mathbf{f}\, d\mathbf{g}\, d\mathbf{u}_f\, d\mathbf{u}_g \\ &\geq \int q(\mathbf{f})q(\mathbf{g}) \log p(\mathbf{y}|\mathbf{f},\mathbf{g})d\mathbf{f}\, d\mathbf{g} - \mathrm{KL}\left(q(\mathbf{u}_f) \,\|\, p(\mathbf{u}_f)\right) - \mathrm{KL}\left(q(\mathbf{u}_g) \,\|\, p(\mathbf{u}_g)\right), \end{aligned} \quad (3)$$

where $q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u}_f)q(\mathbf{u}_f)d\mathbf{u}_f$ and $q(\mathbf{g}) = \int p(\mathbf{g}|\mathbf{u}_g)q(\mathbf{u}_g)d\mathbf{u}_g$, and $\mathrm{KL}\left(p(a) \,\|\, p(b)\right)$ denotes the KL divergence between the two distributions. For Gaussian process priors on the latent functions we recover

$$p(\mathbf{f}|\mathbf{u}_f) = \mathcal{N}\left(\mathbf{f}|\mathbf{K}_{\mathbf{f}\mathbf{u}_f}\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}\mathbf{u}_f, \mathbf{K}_{\mathbf{f}\mathbf{f}} - \mathbf{Q}_{\mathbf{f}\mathbf{f}}\right)$$
$$p(\mathbf{g}|\mathbf{u}_g) = \mathcal{N}\left(\mathbf{g}|\mathbf{K}_{\mathbf{g}\mathbf{u}_g}\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}\mathbf{u}_g, \mathbf{K}_{\mathbf{g}\mathbf{g}} - \mathbf{Q}_{\mathbf{g}\mathbf{g}}\right),$$

where $\mathbf{Q}_{\mathbf{f}\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{u}_f}\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}\mathbf{K}_{\mathbf{u}_f\mathbf{f}}$ and $\mathbf{Q}_{\mathbf{g}\mathbf{g}} = \mathbf{K}_{\mathbf{g}\mathbf{u}_g}\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}\mathbf{K}_{\mathbf{u}_g\mathbf{g}}$. Note that covariances for $\mathbf{f}$ and $\mathbf{g}$, can differ though their inducing input locations, $\mathbf{Z}$, are shared.

We take $q(\mathbf{u}_f)$ and $q(\mathbf{u}_g)$ to be Gaussian distributions with variational parameters, $q(\mathbf{u}_f) = \mathcal{N}(\mathbf{u}_f|\boldsymbol{\mu}_f, \boldsymbol{S}_f)$ and $q(\mathbf{u}_g) = \mathcal{N}(\mathbf{u}_g|\boldsymbol{\mu}_g, \boldsymbol{S}_g)$. Using the properties of multivariate Gaussians this results in tractable integrals for $q(\mathbf{f})$ and $q(\mathbf{g})$,

$$q(\mathbf{f}) = \mathcal{N}\left(\mathbf{f}|\mathbf{K}_{\mathbf{f}\mathbf{u}_f}\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}\boldsymbol{\mu}_f, \mathbf{K}_{\mathbf{f}\mathbf{f}} + \hat{\mathbf{Q}}_{\mathbf{f}\mathbf{f}}\right) \quad (4)$$

$$q(\mathbf{g}) = \mathcal{N}\left(\mathbf{g}|\mathbf{K}_{\mathbf{g}\mathbf{u}_g}\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}\boldsymbol{\mu}_g, \mathbf{K}_{\mathbf{g}\mathbf{g}} + \hat{\mathbf{Q}}_{\mathbf{g}\mathbf{g}}\right), \quad (5)$$

where $\hat{\mathbf{Q}}_{\mathbf{f}\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{u}_f}\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}(\boldsymbol{S}_f - \mathbf{K}_{\mathbf{u}_f\mathbf{u}_f})\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}\mathbf{K}_{\mathbf{u}_f\mathbf{f}}$ and $\hat{\mathbf{Q}}_{\mathbf{g}\mathbf{g}} = \mathbf{K}_{\mathbf{g}\mathbf{u}_g}\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}(\boldsymbol{S}_g - \mathbf{K}_{\mathbf{u}_g\mathbf{u}_g})\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}\mathbf{K}_{\mathbf{u}_g\mathbf{g}}$.

The KL terms in (3) and their derivative can be computed in closed form and are inexpensive as they are divergence between Gaussians. However, an intractable integral, $\int q(\mathbf{f})q(\mathbf{g}) \log p(\mathbf{y}|\mathbf{f},\mathbf{g})d\mathbf{f}\, d\mathbf{g}$, still remains. Since the likelihood factorizes,

$$p(\mathbf{y}|\mathbf{f},\mathbf{g}) = \prod_{i=1}^{n} p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i),$$

the problematic integral in (3) also factorizes across data points, allowing us to use stochastic variational inference [Hensman et al., 2013a, Hoffman et al., 2013],

$$\begin{aligned}&\int q(\mathbf{f})q(\mathbf{g}) \log p(\mathbf{y}|\mathbf{f},\mathbf{g})d\mathbf{f}\, d\mathbf{g} \\ &= \int q(\mathbf{f})q(\mathbf{g}) \log \prod_{i=1}^{n} p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i)d\mathbf{f}\, d\mathbf{g} \\ &= \sum_{i=1}^{n} \int q(\mathbf{f}_i)q(\mathbf{g}_i) \log p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i)d\mathbf{f}_i\, d\mathbf{g}_i. \end{aligned} \quad (6)$$

We are then left with n, b dimensional Gaussian integrals over the log-likelihood,

$$\log p(\mathbf{y}) \geq \sum_{i=1}^{n} \int q(\mathbf{f}_i)q(\mathbf{g}_i) \log p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i)d\mathbf{f}_i\, d\mathbf{g}_i - \mathrm{KL}\left(q(\mathbf{u}_f) \,\|\, p(\mathbf{u}_f)\right) - \mathrm{KL}\left(q(\mathbf{u}_g) \,\|\, p(\mathbf{u}_g)\right). \quad (7)$$

The bound will also hold for any additional number of latent functions by assuming they all factorize in the variational posterior.

The bound decomposes into a sum over data, as such the n input points can be visited in mini-batches, and the gradients and log-likelihood of each mini-batch can be subsequently summed, this operation can be also be parallelized [Gal et al., 2014]. A single mini-batch can instead be visited obtaining a stochastic gradient for use in a stochastic optimization [Hensman et al., 2013a, Hoffman et al., 2013]. This provides the ability to scale to huge datasets.

If the likelihood is Gaussian these integrals are analytic [Lazaro-Gredilla and Titsias, 2011], though the noise variance must be constrained positive via a transformation of the latent function, e.g an exponent. In this case,

$$\begin{aligned}&\int q(\mathbf{f}_i)q(\mathbf{g}_i) \log p(\mathbf{y}_i|\mathbf{f}_i,\mathbf{g}_i)d\mathbf{f}_i\, d\mathbf{g}_i \\ &= \int \mathcal{N}\left(\mathbf{f}_i|\boldsymbol{m}_{f_i}, \boldsymbol{v}_{f_i}\right)\mathcal{N}\left(\mathbf{g}_i|\boldsymbol{m}_{g_i}, \boldsymbol{v}_{g_i}\right) \log \mathcal{N}\left(\mathbf{y}_i|\mathbf{f}_i, e^{\mathbf{g}_i}\right) \\ &= \log \mathcal{N}\left(\mathbf{y}_i|\boldsymbol{m}_{f_i}, e^{\boldsymbol{m}_{g_i} - \frac{\boldsymbol{v}_{g_i}}{2}}\right) - \frac{\boldsymbol{v}_{g_i}}{4} - \frac{\boldsymbol{v}_{f_i}e^{-\boldsymbol{m}_{g_i} + \frac{\boldsymbol{v}_{g_i}}{2}}}{2}\end{aligned}$$

where we define $\boldsymbol{m}_f = \mathbf{K}_{\mathbf{f}\mathbf{u}_f}\mathbf{K}^{-1}_{\mathbf{u}_f\mathbf{u}_f}\boldsymbol{\mu}_f$, $\boldsymbol{v}_f = \mathbf{K}_{\mathbf{f}\mathbf{f}} + \hat{\mathbf{Q}}_{\mathbf{f}\mathbf{f}}$, $\boldsymbol{m}_g = \mathbf{K}_{\mathbf{g}\mathbf{u}_g}\mathbf{K}^{-1}_{\mathbf{u}_g\mathbf{u}_g}\boldsymbol{\mu}_g$, $\boldsymbol{v}_g = \mathbf{K}_{\mathbf{g}\mathbf{g}} + \hat{\mathbf{Q}}_{\mathbf{g}\mathbf{g}}$. $\boldsymbol{v}_{f_i}$ denotes the ith diagonal element of the matrix with $\boldsymbol{v}_f$ along its diagonal. It may be possible in this Gaussian case to find the optimal $q(\mathbf{f})$ such that the bound collapses to that of Lazaro-Gredilla and Titsias [2011], however this would not allow for stochastic optimization. Here we arrive at a sparse extension, where a Gaussian distribution is assumed for the posterior over of $\mathbf{f}$, where as previously $q(\mathbf{f})$ has been collapsed out and could take any form. This sparse extension provides the ability to scale to much larger datasets whilst maintaining a similar variational lower bound.

The model is however not restricted to heteroscedastic Gaussian likelihoods. If the integral (6) and its gradients can be computed in an unbiased way, any factorizing likelihood can be used. This can be seen as a chained Gaussian process. There is no single link function that allows the specification of this model under the modelling assumptions of a generalised linear model. An example that will be revisited in the experiments is the beta distribution, $y_i \sim B(\alpha, \beta)$ where $\alpha, \beta \in \mathbb{R}^+$ and observations $y_i \in (0, 1)$, $\mathbf{x}_i \in \mathbb{R}^q$. Since α, β must maintain positiveness, then can be assigned log GP priors,

$$y_i \sim B(\alpha = e^{f(\mathbf{x}_i)}, \beta = e^{g(\mathbf{x}_i)}), \tag{8}$$

where $f(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_f, k_f(\mathbf{x}, \mathbf{x}'))$ and $g(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_g, k_g(\mathbf{x}, \mathbf{x}'))$. This allows the shape of the beta distribution to change over time, see Supplementary Material and section 4.2.2 for an example plots.

Using the variational bound above, all that is required is to perform a series of n two dimensional quadratures, for both the log-likelihood and its gradients, a relatively simple task and computationally feasible when looking at modest batch sizes. From this example the power and adaptability of the method should be apparent.

A major strength of this method is that performing this integral is the only requirement to implement a new noise model, similarly to [Nguyen and Bonilla, 2014, Hensman et al., 2015]. Further, since a stochastic optimizer is used the gradients do not need to be exact. Our implementations can use off the shelf stochastic optimizer, such as Adagrad [Duchi et al., 2011] or RMSProp [Tieleman and Hinton, 2012]. Further, for many likelihoods some portion of these integrals is analytically tractable, reducing the variance introduced by numerical integration. See supplementary material for an investigation.

3.2 Posterior and Predictive Distributions

Following from (2) it is clear that when the variational lower bound bound has been maximised with respect to the variational parameters, $p(\mathbf{u}_f|\mathbf{y}) \approx q(\mathbf{u}_f)$ and $p(\mathbf{u}_g|\mathbf{y}) \approx q(\mathbf{u}_g)$. The posterior for $p(\mathbf{f}^*|\mathbf{y}^*)$ under this approximation is

$$\begin{aligned} p(\mathbf{f}^*|\mathbf{y}^*) &= \int p(\mathbf{f}^*|\mathbf{x}, \mathbf{f}) p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{u}_f|\mathbf{y}) d\mathbf{f}\, d\mathbf{u}_f \\ &= \int p(\mathbf{f}^*|\mathbf{u}_f) p(\mathbf{u}_f|\mathbf{y}) d\mathbf{u}_f \\ &\approx \int p(\mathbf{f}^*|\mathbf{u}_f) q(\mathbf{u}_f) d\mathbf{u}_f = q(\mathbf{f}^*), \end{aligned}$$

where $q(\mathbf{f}^*)$ and $q(\mathbf{g}^*)$ become similar to (4).

Finally, treating each prediction point independently, the predictive distribution for each data pair $\{(\mathbf{x}_i^*, \mathbf{y}_i^*)\}_{i=1}^{n^*}$ follows as

$$p(\mathbf{y}_i^*|\mathbf{y}_i, \mathbf{x}_i) = \int p(\mathbf{y}_i^*|\mathbf{f}_i^*, \mathbf{g}_i^*) q(\mathbf{f}_i^*) q(\mathbf{g}_i^*) d\mathbf{f}_i^*\, d\mathbf{g}_i^*.$$

This integral is analytically intractable in the general case, but again can be computed using a series of two dimensional quadrature or simple Monte Carlo sampling.

4 Experiments

To evaluate the effectiveness of our chained GP approximations we consider a range of real and synthetic datasets. The performance measure used throughout is the negative log predictive density (NLPD) on held out data, table 1.[1] The results for mean absolute error (MAE) (Supplementary Material) show comparable results between methods. 5-fold cross-validation is used throughout. The non-linear optimization of (hyper-) parameters is subject to local minima, as such multiple runs were performed on each fold with a range of parameter initialisations. The solution obtaining the highest log-likelihood on the training data of each fold was retained. Automatic relevance determination exponentiated quadratic kernels are used throughout allowing one lengthscale per input dimension, in addition to a bias kernel.[2]. In all experiments 100 inducing points were used and their locations were optimized with respect to the lower bound of the log marginal likelihood following [Titsias, 2009].

4.1 Heteroscedastic Gaussian

In our introduction we related our ideas to heteroscedastic GPs. We will first use our approximations to show how the addition of input dependent noise to a Gaussian process regression model effects performance, compared with a sparse Gaussian process model [Titsias, 2009]. Performance is shown to improve as more data is provided as would be expected, making it clear that both models can scale with data, though the new model is more flexible when handling the distributions tails. A sparse Gaussian process with Gaussian likelihood is chosen in these experiments as a baseline, as a non-sparse Gaussian process cannot scale to the size of all the experiments.

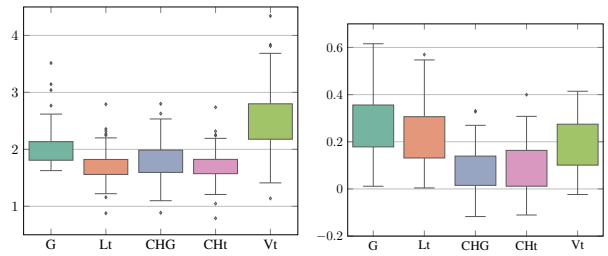
The Elevator1000 uses a subset of $1,000$ of the Elevator dataset . In this data the heteroscedastic model (Chained GP) offers considerable improvement in terms of negative log predictive density (NLPD) over the sparse GP (Table 1. Our second experiment with the Gaussian likelihood, Elevator10000, examines scaling of the model. Here a subset of $10,000$ data points of the Elevator dataset are used, and

[1] Data used in the experiments can be downloaded via the pods package: `https://github.com/sods/ods`

[2] Code is publically available at: `https://github.com/SheffieldML/ChainedGP`

Data	NLPD				
	G	CHG	Lt	Vt	CHt
elevators1000	0.39 ± 0.13	0.1 ± 0.01	NA	NA	NA
elevators10000	0.07 ± 0.01	0.03 ± 0.02	NA	NA	NA
motorCorrupt	2.04 ± 0.06	1.79 ± 0.05	1.73 ± 0.05	2.52 ± 0.09	1.7 ± 0.05
boston	0.27 ± 0.02	0.09 ± 0.01	0.23 ± 0.02	0.19 ± 0.02	0.09 ± 0.02

Table 1: Results NLPD over 5 cross-validation folds with 10 replicates each. Models shown in comparison are sparse Gaussian (G), chained heteroscedastic Gaussian (CHG), Student-t Laplace approximation (Lt), Student-t VB approximation (Vt), and chained heteroscedastic Student-t (CHt).



(a) NLPD motor (b) NLPD Boston

Figure 1: a) NLPD on corrupt motorcycle dataset. b) NLPD of Boston housing dataset. In NLPD lower is better, models shown in comparison are sparse Gaussian (G), Student-t Laplace approximation (Lt), Student-t VB approximation (Vt), chained heteroscedastic Gaussian (CHG), and chained heteroscedastic Student-t (CHt). Boxplots show the variation over 5 folds.

performance is improved as expected. Previous models for heteroscedastic Gaussian process models cannot scale, the chained GP can implement the heteroscedastic setting and scale.

4.2 Non-Gaussian Heteroscedastic likelihoods

One of the major strengths of the approximation over pure scalability, is the ability to use more general non-Gaussian likelihoods. In this section we will investigate this flexibility by performing inference with non-standard likelihoods. This allows models to be specified that correspond to the modellers belief about the data in a flexible way.

We first investigate an extension of the Student-t likelihood that endows it with an input-dependent scale parameter. This is straightforward in the chained GP paradigm.

The corrupt motorcycle dataset is an artificial modification to the benchmark motorcycle dataset [Silverman, 1985] and shows the models capabilities more clearly. The original motorcycle dataset has had 25 of its data randomly corrupted with Gaussian noise of $\mathcal{N}(0, 3)$, simulating spurious accelerometer readings. We hope that our method will be robust and ignore such outlying values. An input-dependent mean, μ, is set alongside an input dependent scale which must be positive, σ. A constant degrees of freedom parameter ν, is initalized to 4.0 and then is optimized to its MAP solution.

$$y_i \sim St(\mu = f(\mathbf{x}_i), \sigma^2 = e^{g(\mathbf{x}_i)}, \nu) \tag{9}$$

where $f(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_f, k_f(\mathbf{x}, \mathbf{x}'))$ and $g(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_g, k_g(\mathbf{x}, \mathbf{x}'))$. This provides a heteroscedastic extension to the Student-t likelihood. We compare the model with a Gaussian process with homogeneous Student-t likelihood, approximated variationally [Hensman et al., 2015] and the Laplace approximation. Figure 2 shows the improved quality of the error bars with the chained heteroscedastic Student-t model. Learning a model with heavy tails allows outliers to be ignored, and so its input dependent variance can be collapsed around just points close to the underlying function, which in this case is known to be well modelled with a heteroscedastic Gaussian process [Lazaro-Gredilla and Titsias, 2011, Goldberg et al., 1998]. It is also interesting to note the heteroscedastic Gaussian's performance, although not able to completely ignore outliers the model has learnt a very short length-scale. This renders the prior over the scale parameter independent across the data, meaning that the resulting likelihood is more akin to a scale-mixture of Gaussians (which endows appropriate robustness characteristics). The main difference is that the scale-mixture is based on a log-Gaussian prior, as opposed to the Student-t which is based on an inverse Gamma.

Figure 1 shows the NLPD on the corrupt motorcycle dataset and Boston housing dataset. The Boston housing dataset shows the median house prices throughout the Boston area, quantified by 506 data points, with 13 explanatory input variables [Kuß, 2006]. We find that the chained heteroscedastic Gaussian process model already outperforms the Student-t model on this dataset, and the additional ability to use heavier tails in the chained Student-t is not used. This ability to regress back to an already powerful model is a useful property of the chained Student-t model.

4.2.1 Survival Analysis

Survival analysis focuses on the analysis of time-to-event data. This data arises frequently in clinical trials, though it is also commonly found in failure tests within engineering. In these settings it is common to observe censoring.

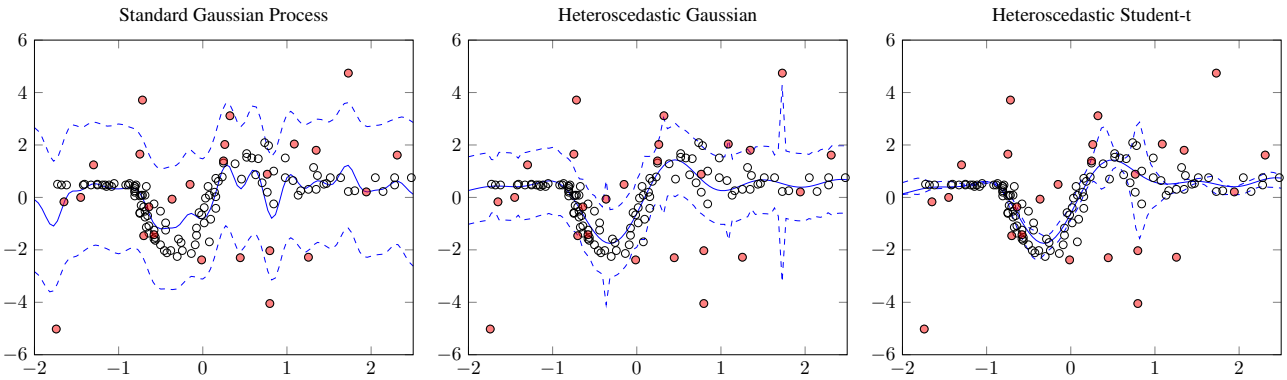


Figure 2: Corrupted motorcycle dataset, fitted with a Gaussian process model with a Gaussian likelihood, a Gaussian process with input dependent noise (heteroscedastic) with a Gaussian likelihood, and a Gaussian process with Student-t likelihood, with an input dependent shape parameter. The mean is shown in solid and the variance is shown as dotted

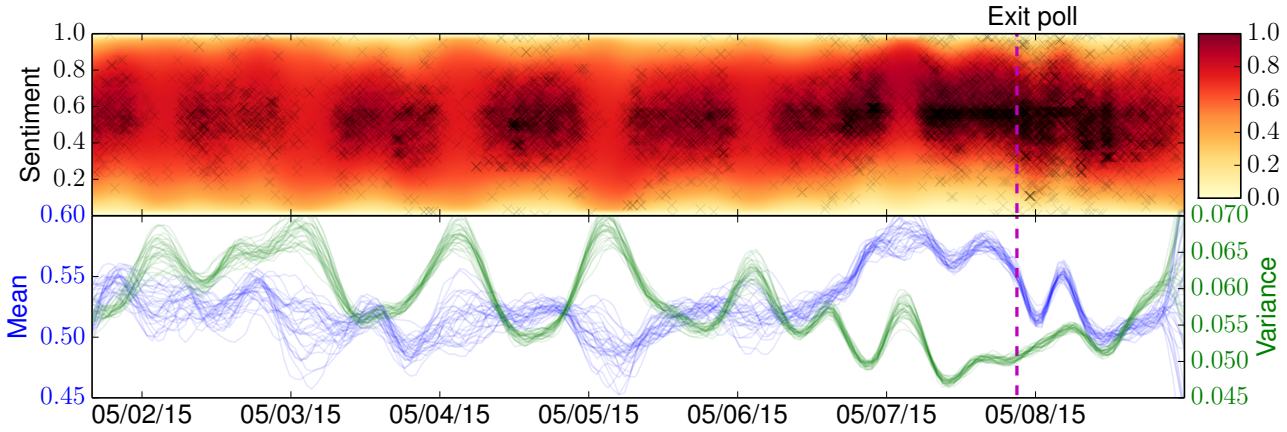


Figure 3: Twitter sentiment from the UK general election modelled using a heteroscedastic beta distribution. The timing of the exit poll is marked and is followed by a night of tweets as election counts come in. Other night time periods have a reduced volume of tweets and a corresponding increase in sentiment variance. Ticks on the x-axis indicate midnight.

Censoring occurs when an event is only observed to exist between two times, but no further information is available. For right-censoring, the most common type of censoring, the event time $T \in [t, \infty)$.

A common model to analyse this type of data is an *accelerated failure time* model. This suggests that the distribution of when an random event, T, may occur, is multiplicatively effected by some function of the covariates, $f(\mathbf{x})$, thus accelerating or retarding time; akin to notion of dog years. In a generalized linear model we may write this as $\log T = \log T_0 + \log f(\mathbf{x})$, where T is the random variable for failure time of the individual with covariates $\mathbf{x}$, and T_0 follows a parametric distribution describing a non-accelerated failure time.

To account for censoring the cumulative distribution needs to be computable and event times are restricted to be positive. A common parametric distribution for T_0 that fulfills these restrictions is the log-logistic distribution, with the median being some function of the covariates, $f(\mathbf{x})$. This however is restrictive as the *shape* of failure time distribution is assumed to be the same for all patients. We relax this assumption by allowing the shape parameter of the log-logistic distribution to vary with response to the input,

$$y_i \sim LL(\alpha = e^{f(\mathbf{x}_i)}, \beta = e^{g(\mathbf{x}_i)}),$$

where $f(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_f, k_f(\mathbf{x}, \mathbf{x}'))$ and $g(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_g, k_g(\mathbf{x}, \mathbf{x}'))$. This allows both skewed unimodal and exponential shaped distributions for the failure time distribution depending on the individual, as shown in Figure 4. Again there is no associated link-function in this case, and the model can be modelled as a chained-survival model.

Table 2 shows the models performance a real and synthetic datasets. The leukemia dataset [Henderson et al., 2002] contains censored event times for 1043 leukemia patients and is known to have non-linear responses certain covariates [Gelman et al., 2013]. We find little advantage from using the chained-survival model, but as usual the model is

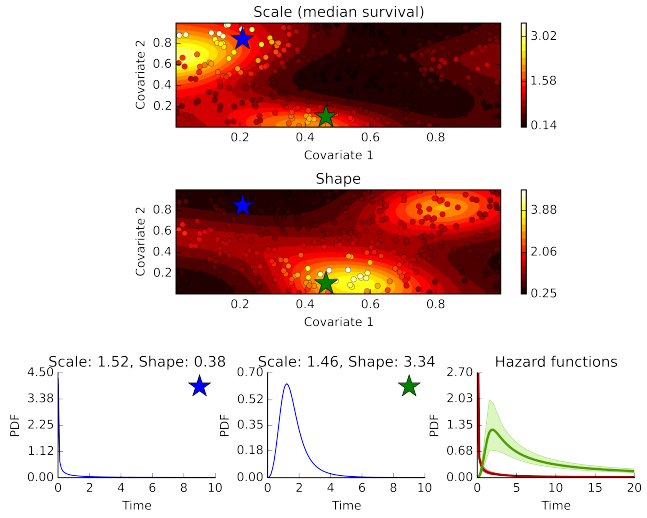


Figure 4: Resulting model on synthetic survival dataset. Shows variation of median survival time and shape of log-logistic distribution, in response to differing covariate information. Background colour shows the chained-survivals predictions, coloured dots show ground truth. Lower figures show associated failure time distributions and hazards for two different synthetic patients. Shapes can be both unimodal or exponential.

Data	NLPD			
	G	LSurv	VSurv	CHSurv
leuk	$4.03 \pm .08$	$1.57 \pm .01$	$1.57 \pm .01$	$1.56 \pm .01$
Surv	$5.45 \pm .06$	$2.52 \pm .02$	$2.52 \pm .02$	$2.16 \pm .02$

Table 2: Results NLPD over 5 cross-validation folds with 10 replicates each. Models shown in comparison are sparse Gaussian (G), survival Laplace approximation (LSurv), survival VB approximation (VSurv), chained heteroscedastic survival (CHSurv).

robust such that performance isn't degraded in this case. We additionally show the results on a synthetic dataset where the shape parameter is known to vary with response to the input, in this case an increase in performance is seen. See Appendix A.5 for more details on the model and synthetic dataset.

4.2.2 Twitter Sentiment Analysis in the UK Election

The final experiment shows the adaptability of the model even further, on a novel dataset and with a novel heteroscedastic model. We consider sentiment in the UK general election, focussing on tweets tagged as supporters of the Labour party. We used a sentiment analysis tagging system[3] to evaluate the positiveness of 105,396 tweets containing hashtags relating to recent the major political parties, over the run in to the UK 2015 general election. We are interested in modeling the distribution of positive sentiment as a function of time. The sentiment value is constrained to be to be between zero and one, and we do not believe the distribution of tweets through time to be necessarily unimodal. A natural likelihood to use in this case is the beta likelihood. This allows us to accommodate bathtub shaped distributions, indicating tweets are either extremely positive or extremely negative. We then allow the distribution over tweets to be heterogenous throughout time by using Gaussian process models for each parameter of the beta distribution, $y_i \sim B(\alpha = e^{f(\mathbf{x}_i)}, \beta = e^{g(\mathbf{x}_i)})$, where $f(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_f, k_f(\mathbf{x}, \mathbf{x}'))$ and $g(\mathbf{x}) = \mathcal{GP}(\boldsymbol{\mu}_g, k_g(\mathbf{x}, \mathbf{x}'))$.

The upper section of Figure 3 shows the data and the probability of each sentiment value throughout time. The lower part shows the corresponding mean and variance functions induced by the above parameterization. This year's general election was particularly interesting: polls throughout the election showed it to be a close race between the two major parties, Conservative and Labour. But at the end of polling an exit poll was released that predicted an outright win for the Conservatives. This exit poll proved accurate and is associated with a corresponding dip in the sentiment of the tweets. Other interesting aspects of the analysis include the reduction in number of tweets during the night and the corresponding increase in the variance of our estimates.

4.2.3 Decomposition of Poisson Processes

The intensity, $\lambda(x)$, of a Poisson process can be modelled as the product of two positive latent functions, $\exp(f(x))$ and $\exp(g(x))$, as a generalrised linear model,

$$\log(\lambda) = f(x) + g(x)$$
$$y \sim \text{Poisson}(\lambda = \exp(f + g) = \exp(f(x))\exp(g(x))),$$

using a *log* link function.

Instead imagine we form a new process by combining two different underlying Poisson processes through addition. The superposition property of Poissons means that the resulting process is also Poisson with rates given by the sum of the underlying rates.

To model this via a Gaussian process we have to assume that the intensity of the resulting Poisson, $\lambda(x)$ is a *sum* of two positive functions $\exp(f(x))$ and $\exp(g(x))$,

$$y \sim \text{Poisson}(\lambda = \exp(f(x)) + \exp(g(x))), \qquad (10)$$

there is no link function representation for this model, it takes the form of a chained-GP.

Focusing purely on the generative model of the data, the lack of an link function does not present an issue. Figure 5 shows a simple demonstration of the idea in a simulated data set.

Using an additive model for the rate rather than a multiplicative model for counting processes has been discussed

[3] Available from `https://www.twinword.com/`

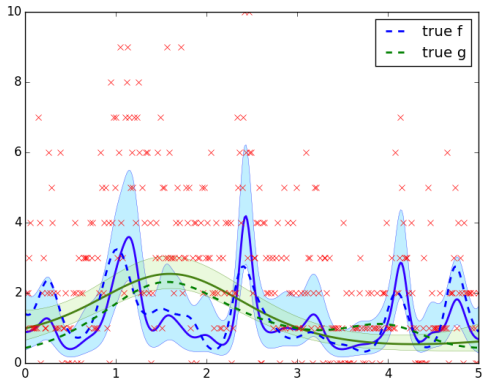


Figure 5: Even with 350 data we can start to see the differentiation of the addition of a long lengthscale positive process and a short lengthscale positive process. Red crosses denote observations, dotted lines are the true latent functions generating the data using Eq (10), the solid line and associated error bars are the approximate posterior predictions, $q(\mathbf{f}^*), q(\mathbf{g}^*)$, of the latent processes.

previously in the context of linear models for survival analysis, with promising results Lin and Ying [1995].

To illustrate the model on real data we considered homicide data in Chicago. Taking data from `http://homicides.redeyechicago.com/` (see also Linderman and Adams [2014]) we aggregated data into three months periods by zip code. We considered an additive Poisson process with a particular structure for the covariance functions. We constructed a rate of the form:

$$\Lambda(x,t) = \lambda_1(x)\mu_1(t) + \lambda_2(x)\mu_2(t)$$

where $\lambda_1(x) = \exp(f_1(x))$, $\lambda_2(x) = \exp(g_1(x))$, $\mu_1(t) = \exp(f_2(t))$ and $\mu_2(t) = \exp(g_2(t))$ where $f_1(x)$, $g_1(x)$ are spatial GPs and $f_2(t)$ and $g_2(t)$ are temporal GPs. The overall rate decomposes into two separable rate functions, but the overall rate function is not separable. We have a sum of separable [Álvarez et al., 2012] rate functions. This structure allows us to decompose the homicide map into separate spatial maps that each evolve at different time rates. We selected one spatial map with a length scale of 0.04 and one spatial map with a length scale of 0.09. The time scales and variances of the temporal rate functions were optimized by maximum likelihood. The results are shown in Figure 6. The long length scale process hardly fluctuates across time, whereas the short lengthscale process, which represents more localized homicide activity, fluctuates across the seasons with scaled increases of around 1.25 deaths per month per zip code. This decomposition is possible and interpretable due to the structured underlying nature of the GPs inside the chained model.

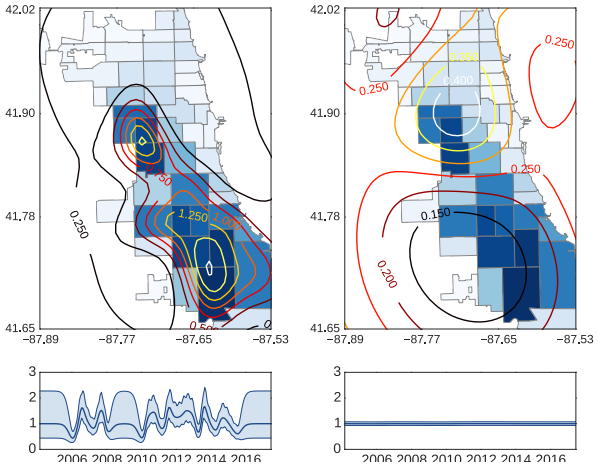


Figure 6: Homicide rate maps for Chicago. The short length scale spatial process, $\lambda_1(x)$ (above-left) is multiplied in the model by a temporal process, $\mu_1(t)$ (below-left) which fluctuates with passing seasons. Contours of spatial process are plotted as deaths per month per zip code area. Error bars on temporal processes are at 5th and 95th percentile. The longer length scale spatial process, $\lambda_2(x)$ (above-right) has been modeled with little to no fluctuation temporally $\mu_2(t)$ (below-right).

5 Conclusions

We have introduced "Chained Gaussian Process" models. They allow us to make predictions which are based on a non-linear combination of underlying latent functions. This gives a far more flexible formalism than the generalized linear models that are classically applied in this domain.

Chained Gaussian processes are a general formalism and therefore are intractable in the base case. We derived an approximation framework that is applicable for any factorized likelihood. For the cases we considered, involving two latent functions, the approximation made use of two dimensional Gauss-Hermite quadrature. We speculated that when the idea is extended to higher numbers of latent functions it may be necessary to resort to Monte Carlo sampling.

Our approximation is highly scalable through the use of stochastic variational inference. This enables the full range of standard stochastic optimizers to be applied in the framework.

Acknowledgments

AS was supported by a University of Sheffield, Faculty Scholarship, JH was supported by a MRC fellowship. The authors also thank Amazon for donated AWS compute time and the anonymous reviewers for their comments.

References

Ryan Prescott Adams and Oliver Stegle. Gaussian process product models for nonparametric nonstationarity. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1–8, 2008.

Mauricio Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. doi: 10.1561/2200000036.

Amir Dezfouli and Edwin V Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1414–1422. Curran Associates, Inc., 2015.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435.

Yarin Gal, Mark van der Wilk, and Carl E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, Cambridge, MA, 2014.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis, Third Edition*. CRC Press, November 2013. ISBN 9781439840955.

Paul W. Goldberg, Christopher K. I. Williams, and Christopher M. Bishop. Regression with input-dependent noise: A Gaussian process treatment. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 493–499, Cambridge, MA, 1998. MIT Press.

R. Henderson, S. Shimakura, and Gorst D. Modeling spatial variation in leukemia survival data. *Journal of the American Statistical Association*, 97:965–972, 2002.

James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. In Ann Nicholson and Padhraic Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013a.

James Hensman, Neil D. Lawrence, and Magnus Rattray. Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 14(252), 2013b. doi: doi:10.1186/1471-2105-14-252.

James Hensman, Alexander G D G Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *In 18th International Conference on Artificial Intelligence and Statistics*, pages 1–9, San Diego, California, USA, May 2015.

Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 837–845. Curran Associates, Inc., 2014.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust Gaussian process regression with a Student-t likelihood. *J. Mach. Learn. Res.*, 12:3227–3257, November 2011. ISSN 1532-4435.

D. P. Kingma and M. Welling. Stochastic gradient VB and the variational auto-encoder. In *2nd International Conference on Learning Representations (ICLR)*, Banff, 2014.

Malte Kuß. *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. PhD thesis, TU Darmstadt, April 2006.

Miguel Lazaro-Gredilla and Michalis Titsias. Variational heteroscedastic Gaussian process regression. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 841–848, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

D. Y. Lin and Zhiliang Ying. Semiparametric analysis of general additive-multiplicative hazard models for counting processes. *The Annals of Statistics*, 23(5):1712–1734, 10 1995. doi: 10.1214/aos/1176324320.

Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *ICML*, 2014.

John Nelder and Robert Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, A*, 135(3), 1972. doi: 10.2307/2344614.

Trung V Nguyen and Edwin V Bonilla. Automated variational inference for Gaussian process models. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1404–1412. Curran Associates, Inc., 2014.

Manfred Opper and Cedric Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–792, 2009.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and variational inference in deep latent Gaussian models. Technical report, 2014.

B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society, B*, 47(1):1–52, 1985.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Yair Weiss, Bernhard Schölkopf, and John C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

Edward Snelson, Carl Edward Rasmussen, and Zoubin Ghahramani. Warped Gaussian processes. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Cambridge, MA, 2004. MIT Press.

T. Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. In: COURSERA: Neural Networks for Machine Learning, 2012.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.

Ville Tolvanen, Pasi Jylänki, and Aki Vehtari. Expectation propagation for nonstationary heteroscedastic Gaussian process regression. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop*, 2014.

Richard E. Turner and Maneesh Sahani. Demodulation as probabilistic inference. *IEEE Transactions on Audio, Speech, and Language Processing*, 19:2398–2411, 2011.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14(1):1175–1179, 2013. `http://mloss.org/software/view/451/`.